CLAIMS

What is claimed is:

1. A system for using and protecting access to a master cryptographic key, comprising:

non-volatile storage;

a system initialization process that:

reads the master key from the non-volatile storage during a system initialization process;

writes a sensitive value derived from the master key to a hidden storage location; and

disables access to the non-volatile storage by any program running in the system until the next start of system initialization process;

means to prevent access to the hidden storage location by programs running in the normal operating mode of the system; and

means to allow access to the hidden storage location by a program running in a restricted operating mode of the system.

2. The system recited in Claim 1 wherein the sensitive data is the master key.

3. The system recited in Claim 1 wherein the sensitive data is derived from the master key.

4. The system recited in Claim 3 wherein the sensitive data is a second key retrieved from encrypted data stored on disk, where the stored data is encrypted with the master key.

5. The system recited in Claim 1 wherein software in BIOS ROM controls the system during the system initialization process that begins in response to a power-on or reset signal.

6. The system recited in Claim 1 wherein

the non-volatile storage is non-volatile random access memory with read and write access controlled by a latch;

the latch is opened at the start of system initialization process due to a hardware function responding to a power-on or reset event, thereby enabling system access to the non-volatile random access memory; and

the latch is closed during the system initialization process, thereby denying system access to the non-volatile random access memory until the next start of system initialization.

7. The system recited in Claims 1 wherein

the hidden storage is system management random access memory which is not accessible by any program running in the normal operating mode of the system; and

the restricted operating mode is a System Management Mode in which access to system management random access memory is permitted.

8. The system recited in Claims 1 wherein

the hidden storage is restricted for access by the operating system only, and is not accessible by any application program that runs in the normal operating mode of the system; and

the restricted operating mode is controlled by a CPU protection ring reserved for use by operating system software.

9. A system for hiding a master cryptographic key in storage, comprising

power-on software that:

reads a master key from non-volatile storage;

closes access to the non-volatile storage such that access does not become available again until the next system reset; and

writes sensitive data derived from the master key to a hidden address space; and

wherein only a program that runs in a restricted operational mode of the system has access to the sensitive data in the hidden address space.

10. A method of controlling read and write access to data to an application by restricting the availability of a cryptographic key to an application, the method comprising:

a master key;

an application container that holds a sealed or unsealed form of the data that the application wants to access;

a cryptographic gatekeeping module that performs a cryptographic digest of a portion of the bytes that make up the calling application to compute a cryptographic transformation; and

a cryptographic processing module that includes integrity-checking that examines the application container and cryptographic transformation, and the master key to determine if the application is allowed to unseal the data in the given application container, or when sealing the data modifies it to add the integrity check information.

11. The method recited in Claim 10 wherein a privacy method performed by the cryptographic processing module that decrypts the data in the application container using a key derived from at least the master key and cryptographic transformation.

12. The method recited in Claim 10 further including a privacy method performed by the cryptographic processing module that encrypts the data in the application container using a key derived from at least the master key and cryptographic transformation.

13. The method recited in Claim 12 wherein the privacy method adds to the application container the cryptographic transformation before the encryption is performed.

14. A method of controlling access to data to an application by restricting the availability of a cryptographic key to the application on a specific device, comprising:

a key known to a cryptographic processing module;

an application container data structure that contains a cryptographically sealed form of the data that the application wants to access;

a cryptographic gatekeeping function that

intercepts all access between application-level programs and the cryptographic processing module;

includes a means to examine a portion of the bytes of an executable in-memory image of a program that is attempting to access cryptographic services or data; and

computes a cryptographic digest of a portion of the bytes of in-memory image of the calling application to compute the cryptographic transformation of the application; and

an integrity-check method performed by the cryptographic processing module that examines the application container data structure and cryptographic transformation,

and the master key to determine if the application is allowed to unseal the data in the given application container data structure, or when sealing the data modifies it to add the integrity check information.

15. The method recited in Claim 14 further comprising a privacy method performed by the cryptographic processing module that encrypts or decrypts the data in the application container data structure using a key derived from at least the master key and cryptographic transformation and when data is encrypted it optionally adds to the application container data structure the cryptographic transformation before the encryption is performed.

16. The method recited in Claim 14 wherein the cryptographic gatekeeping function is concurrently or previously given an authorization buffer that specifies the allowed operations for the application and the cryptographic gatekeeping function confirms that the request operation is allowed.

17. The method recited in Claims 14 wherein the integrity-check method includes the steps of deriving a cryptographic variable from the cryptographic transformation and the master key, or of deriving a second cryptographic variable from the cryptographic transformation, the master key and a cryptographic variable chosen by a component of an application, and this derived key is used to check a message authentication code that is stored in the application container data structure.

18. The method recited in Claims 14 wherein the integrity-check method includes decrypting a portion of the application container data structure using a key derived from the master key and comparing a portion of the resulting value to a portion of the cryptographic transformation, and allowing the access if the two portions are the same.

19. The method recited in Claims 14 wherein the privacy step includes the steps of deriving a cryptographic variable from the cryptographic transformation and the master key and optionally other information, or of deriving a second cryptographic variable from the cryptographic transformation and the master key and a cryptographic variable chosen by a component of an application and optionally other information, and this derived key is used to decrypt or encrypt a portion of the application container data structure.

20. The method recited in Claim 19 wherein the key derivation is performed with one or more applications of the MD5 or SHA1 or SHA-256 hash functions by concatenating the dependant values in some order.

21. The method recited in Claims 14 wherein a portion of the cryptographic processing module executes during an system management interrupt.

22. A method for authenticating an identified application on an identified device to another computing machine comprising an authentication server with the help of another computing machine comprising a device authority, the method comprising:

an enrollment process that includes the steps of:

a) a first cryptographic operation performed during a system management interruption (SMI) on the device producing a result that is sent to the device authority, and

b) a second cryptographic operation performed during an SMI interrupt on the device processing a value generated by the device authority that is received by the device;

a registration process that includes the steps of:

a) a first cryptographic operation performed during an SMI interruption on the Device producing a result that is sent to the authentication server,

b) a second cryptographic operation performed by the authentication server producing a cryptographic variable that is stored for use during the authentication method, and

c) an optional third cryptographic operation performed during an SMI interrupt on the device processing a value generated by the authentication server that is received by the device;

an authentication process that includes the steps of:

a) a first cryptographic operation performed during an SMI interruption on the device producing authentication data that is sent to the authentication server, and

b) a second cryptographic operation performed by the authentication server on the authentication data received from the device using at least the cryptographic variable stored during the registration method to determine the result of the authentication.

23. A method for authenticating an identified application on an identified device, or for providing a second factor for identifying a user of the identified device to another computing machine comprising a PASS server, the method comprising:

an application that

a) performs an enrollment method involving communication with a device authority and an authentication server to create an application container data structure on the device, wherein the application container data structure is cryptographically associated with the application; and

b) stores credential information, and

wherein the authentication server stores a cryptographic variable for the application container data structure;

an application running on the identified device that performs an authentication method including the steps of

a) unsealing the application container data structure that stores the credentials,

b) modifying the credentials;

c) resealing the application container data structure;

d) sending identifying information and at least a portion of the resealed AppContainer to the authentication server;

wherein at least part of the resealing operation takes place during an SMI on the same CPU that executes the code of the application; and

wherein the authentication server

a) receives the identifying information and at least a portion of the application container data structure,

b) uses the identifying information to lookup or compute a cryptographic variable to unseal the application container data structure,

c) if the unsealed application container has acceptable values then the specific application on a specific device is considered to be authenticated; and

d) stores a key associated with the application container data structure.

24. A method for creating and utilizing one or more virtual tokens on a device for the purpose of authentication, privacy, integrity, authorization, auditing, or digital rights management, the method comprising:

an application for each kind of virtual token;

an application container for each virtual token of a specific kind;

a cryptographic gatekeeping component that computes an cryptographic transformation of a calling application that is requesting cryptographic services of a cryptographic processing component;

wherein the cryptographic gatekeeping component knows one or more long-lived symmetric keys;

wherein the cryptographic processing component is accessed via the CryptoGate component;

wherein the cryptographic processing component knows one or more long-lived symmetric keys and one or more long-lived public keys; and

wherein the cryptographic processing component performs cryptographic sealing and unsealing of application container data structures, where a portion of the cryptographic operations are performed during a system management interrupt (SMI);

wherein the cryptographic processing component checks the integrity of the calling application by checking a digital signature of a portion of the application's code or static data, using a public key that has been loaded into the CryptoEngine and a cryptographic transformation value;

wherein the cryptographic transformation value includes a recently computed cryptographic hash of a portion of the calling application's in-memory image;

wherein the cryptographic gatekeeping and cryptographic processing component

a) derive a key for unsealing the application container data structure from the master key and cryptographic transformation,

b) use the derived key to check the message authentication code on the application container data structure, and returns an error if the message authentication code is correct, and

c) use the derived key to decrypt the data in the application container data structure and return it to the application.

25. A method of securely associating a private key with an application associated with a device, comprising:

creating an application container that contains private keys secured by a symmetric key associated with the device.